

Please type a plus sign (+) inside this box →

+

PTO/SB/05 (08-00)

Approved for use through 10/31/2002 OMB 0651-0032

U S Patent and Trademark Office U S DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

**UTILITY  
PATENT APPLICATION  
TRANSMITTAL**

Attorney Docket No. 19312 0005

First Inventor or Application Identifier Ed Anuff

Title METHOD AND SYSTEM FOR PROVIDING REMOTE PORTAL  
SERVICE MODULES

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Express Mail Label No.

**APPLICATION ELEMENTS**

See MPEP chapter 600 concerning design patent application contents.

**ADDRESS TO**Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)  
(Submit an original and a duplicate for fee processing)
2. ☒ Applicant claims small entity status.  
See 37 CFR 1.27.
3. ☒ Specification [Total Pages 43]  
(preferred arrangement set forth below)  
- Descriptive title of the Invention  
- Cross References to Related Applications  
- Statement Regarding Fed sponsored R & D  
- Background of the Invention  
- Brief Summary of the Invention  
- Brief Description of the Drawings (if filed)  
- Detailed Description  
- Claim(s)  
- Abstract of the Disclosure
4. ☒ Drawing(s) (35 U.S.C.113) [Total Sheets 10]  
1
5. Oath or Declaration [Total Pages ]  
a. ☐ Newly executed (original or copy)  
b. ☐ Copy from a prior application (37 CFR 1.63 (d))  
(for a continuation/divisional with Box 17 completed)  
☐ **DELETION OF INVENTOR(S)**  
Signed statement attached deleting inventor(s)  
named in the prior application, see 37 CFR  
1.63(d)(2) and 1.33(b)
6. ☐ Application Data Sheet. See 37 CFR 1.76

7. ☐ CD-ROM or CD-R in duplicate, large table or  
Computer Program (Appendix)
- 8 Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)
- a. ☐ Computer Readable Form (CRF)  
b. Specification Sequence Listing on:  
i. ☐ CD-ROM or CD-R (2 copies); or  
ii. ☐ paper  
c. ☐ Statements verifying identity of above copies

**ACCOMPANYING APPLICATIONS PARTS**

- 9 ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R. § 3.73(b) Statement ☐ Power of  
(when there is an assignee) Attorney
- 11 ☐ English Translation Document (if applicable)
12. ☐ Information Disclosure ☐ Copies of IDS  
Statement (IDS)/PTO-1449 Citations
13. ☐ Preliminary Amendment
- 14 ☒ Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)
- 16 ☐ Other: Fee transmittal

17. If a **CONTINUING APPLICATION**, check appropriate box, and supply the requisite information below and in a preliminary amendment,  
or in an Application Data Sheet under 37 CFR 1.76:

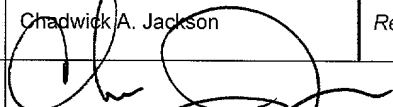
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. \_\_\_\_ /  
Prior application information Examiner \_\_\_\_ Group / Art Unit

For **CONTINUATION** or **DIVISIONAL APPS** only: The entire disclosure of the prior application, from which an oath or declaration is supplied  
under Box 5b, is considered a part of the disclosure of the accompanying or divisional application and is hereby incorporated by reference.  
The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

**17. CORRESPONDENCE ADDRESS**

☒ Customer Number or Bar Code Label 23517 or ☐ Correspondence address below  
(Insert Customer No. or Attach bar code label here)

Name	Edward Pennington					
	Swidler Berlin Shereff Friedman, LLP					
Address	3000 K Street, N.W.					
	Suite 300					
City	Washington	State	D.C.	Zip Code	20007-5116	
Country	USA	Telephone	202 424-7500	Fax	202 424-7647	

Name (Print/Type)	Chadwick A. Jackson	Registration No. (Attorney/Agent)	46,495
Signature		Date	November 22, 2000

Burden Hour Statement This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any  
comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark  
Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for  
Patents, Box Patent Application, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

# FEE TRANSMITTAL for FY 2001

Patent fees are subject to annual revision.

Complete if Known

Application Number --  
Filing Date November 22, 2000  
First Named Inventor Ed Anuff  
Examiner Name TBA  
Group / Art Unit TBA

TOTAL AMOUNT OF PAYMENT (\$) 1303

Attorney Docket No. 19312 0005

jc918 U.S. PTO  
09/717299  
11/22/00

## METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to

Deposit  
Account  
Number

19-5127, Order No 19312 0005

Deposit  
Account  
Name

Swdler Berlin Shereff Friedman, LLP

- ☒ Charge Any Additional Fee Required  
Under 37 CFR 1.16 and 1.17  
☒ Applicant claims small entity status  
See 37 CFR 1.27

2. ☐ Payment Enclosed

☐ Check ☐ Credit card ☐ Money Order ☒ Other

## FEE CALCULATION

### 1. BASIC FILING FEE

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description	Fee Paid
101	710	201	355	Utility filing fee	355
106	320	206	160	Design filing fee	
107	490	207	245	Plant filing fee	
108	710	208	355	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1)

(\$ 355)

### 2. EXTRA CLAIM FEES

				Extra Claims		Fee from below		Fee Paid
Total Claims	112	-20**	=	92	X	9	=	828
Independent Claims	6	-3**	=	3	X	40	=	120
Multiple Dependent					X		=	0

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description
103	18	203	9	Claims in excess of 20
102	80	202	40	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim, if not paid
109	80	209	40	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2)

(\$ 948)

## FEE CALCULATION (continued)

Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	390	216	195	Extension for reply within second month	
117	890	217	445	Extension for reply within third month	
118	1,390	218	695	Extension for reply within fourth month	
128	1,890	228	945	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,240	241	620	Petition to revive - unintentional	
142	1,240	242	620	Utility issue fee (or reissue)	
143	440	243	220	Design issue fee	
144	600	244	300	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	130	123	130	Petitions related to provisional applications	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	710	246	355	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	710	249	355	For each additional invention to be examined (37 CFR § 1.129(b))	
179	710	279	355	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) \_\_\_\_\_

\*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

(\$ 0)

\*\*or number previously paid, if greater. For Reissues, see above

## SUBMITTED BY

Complete (if applicable)

Name (Print/Type) Chadwick A. Jackson Registration No. Attorney/Agent 46,495 Telephone 202 424 7500  
Signature [Signature] Date November 22, 2000

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

## **METHOD AND SYSTEM FOR PROVIDING REMOTE PORTAL SERVICE MODULES**

### **CROSS-REFERENCE TO RELATED APPLICATION**

5           This application is related to application having Ser. No. 09/573,226, filed on  
May 19, 2000, now co-pending.

### **FIELD OF THE INVENTION**

10           The present invention relates generally to the subject of remote applications.  
More particularly, the present invention pertains to providing remote services on a  
portal.

### **BACKGROUND OF THE INVENTION**

15           Traditionally, syndication of services, such as personalized content and  
applications, onto portals has been achieved only through the construction of custom-  
built modules. These custom-built modules provide for the display and administration  
of services hosted on a remote system. While custom built modules have solved the  
problem of syndicating services, their implementation for syndicating personalized  
content and applications negatively impacts the overall development and improvement  
20   of portal framework technology.

A drawback of implementing custom-built modules to syndicate personalized content and applications is delivering upgrades of portal framework technology to customers. Upgrading portal framework technology includes installing new modules on all customer portals that implement the old version of the module. Installing new modules onto customer portals can be very challenging. A primary challenge of installing new modules onto customer portals is that the syndicator must know the users of an old version of a module as well as the version of the module. Knowledge of the version of a module in use by a user is necessary because a new version of a module may only operate on a newer version of the basic portal framework software. Consequently, to ensure the satisfactory and complete delivery of portal framework technology, delivery must be performed a user at a time.

An additional drawback to implementing custom-built modules to syndicate personalized content and applications is that completing an upgrade can take months. The time it takes to complete an upgrade is generally attributed to the significant number of technical tasks that need to be performed. There are, however, occasions when an upgrade of service to all customers may be necessary in an inflexible and short period of time. Such occasions include 1) the syndicator reselling the content or application of a root syndicator; 2) the contract with the root syndicator may change, with immediate consequences on how the portal displays or manages the application; 3) the root syndicator may discontinue their service, requiring that another root syndicator be found to continue the service; or 4) the root syndicator may change their application such that connectors no longer work. Moreover, a contract often exists

between the syndicator and customers that guarantees the continued existence of services. Consequently, due to the number technical tasks, failure to transparently upgrade can occur, and thus lead to legal and financial repercussions.

Another drawback of implementing custom-built modules to syndicate personalized content and applications is the cost of upgrading portal framework technology. Upgrading portal framework technology includes research, development and delivery of new services or features. Research, development and delivery of new services or features requires the expenditure of significant resources. Consequently, upgrade, replacement and/or removal of services or features can often be costly.

An additional drawback of implementing custom-built modules to syndicate personalized content and applications is the frequency at which upgrades occur. As mentioned above, the cost to upgrade portal framework technology is significant and the delivery of upgrades time consuming. Accordingly, upgrades are often performed only when significant advancements are made. Appropriately, “nice to have” functionality is provided only when an upgrade occurs. Consequently, time to market for upgrades in portal framework technology are lengthy.

A further drawback of implementing custom built modules to syndicate personalized content and applications is the inability to easily search and browse for services of both root and third party syndicators. There is no central resource that discloses available services for incorporation. Accordingly, services offered cannot be easily promoted by developers nor identified or obtained by customers. Consequently, the marketing of third parties services to customers is limited.

An additional drawback of implementing custom built modules to provide syndication of personalized content and applications is the limited number functions provided for integrating and customization. Examples of the functions that cannot be performed by a customer include 1) integrating existing complex web applications  
5 into their portal, while retaining the existing interfaces of the application; 2) developing applications for their portal in languages other than Java; and 3) developing applications for their portal where the majority of the computational effort is done on a remote system, rather than on the portal system itself. Customers generally want the ability to perform one or more of the above functions to improve the scalability and customization of their portal site. Consequently, the customer's  
10 ability to customize their portal site is limited.

Accordingly, there is a need for portal framework technology to allow the simultaneous upgrade, replacement and/or removal of a syndicated service or feature. There is a further need for portal framework technology to include a limited number  
15 of technical challenges that syndicators must overcome. There is a need for portal framework technology to enable customers to easily integrate their existing applications. There is an additional need for portal framework technology to shorten service time to market. There is a need for portal framework technology to allow inexpensive upgrade, replacement and/or removal of services or features. There is a  
20 need for portal framework technology to enabling point and click deployment of services into the portal. There is a need for the portal framework technology to provide a technological solution for third parties to market services to customers.

Lastly, there is a need to develop the portal framework technology on top of open standards, such as SOAP and XML.

### SUMMARY OF THE INVENTION

5           Based on the above and foregoing, it can be appreciated that there presently exists a need in the art for portal framework technology which overcomes the above-described deficiencies.

10           To achieve the above and other features and advantages, the present invention provides a remote portal services framework or system. This system or framework will be referred to as the Remote Portal Services (hereinafter "RPS"). A RPS system easily and quickly incorporates personalized content and applications into a portal hosted by the RPS system. Any personalized content and application implemented based on the RPS system will be referred to herein as a RPS Service. A RPS Service may be enabled to make the RPS Service available for addition to a  
15           portal. An enabled RPS Service may also be disabled to terminate the availability of the RPS Service for addition to a portal. A RPS Service may be enabled and disabled through existing portal interface designed for portal administration. A RPS Service is exposed to a users of the RPS system as a module, without installing any software specific to the RPS Service on a device implementing a portal.

20           In another aspect of the present invention, a RPS Service may be listed, located and selected. A RPS Services Directory may be used to list any RPS

Service available for incorporation throughout a network. A RPS Services Directory may enable a RPS Service to be reviewed and selected for incorporation to device. A RPS Service may be represented on the list by a reference identifier, such as a Uniform Resource Identifier (“URI”). Selection of a reference identifier initiates a communication between the server implementing the portal and the server hosting the selected RPS Service.

In another aspect of the present invention, a RPS system may secure server information. RPS may provide security where a response from a RPS Service to a server that has incorporated the RPS Service includes information about the server hosting the RPS Service. The security can prevent, for example, information from being stolen by eavesdropping on the communication or by pretending to be the hosting server.

In a further aspect of the present invention, a RPS system may protect data stored on, or originating on a server implementing a portal from exposure to other servers. A RPS system does not require that a server implementing a portal expose “private data,” to the server hosting a RPS Service. Private data includes anything that is associated with a user and identifies the user uniquely, such as user profile information. A RPS system inspects a RPS Service before it is deployed to make sure it does not request any data considered private. Moreover, the inspection occurs even when a manual process initiated by the customer generates the data.



In another aspect of the present invention, a RPS Service located outside a firewall of an intranet and installed on a portal within the intranet may be accessed. The access may be provided with or without a proxy server.

In another aspect of the present invention, a RPS Service provides current status messages. A RPS Service may specify to servers implementing a RPS Service to display to customers a message in cases where the RPS Service cannot be reached or is temporarily unavailable. RPS Services may specify a timeout for each view within a service.

The features and advantages of the present invention that offer these capabilities are described in detail hereinafter with reference to the accompanying figures, which illustrate exemplary embodiments thereof.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

The details of the present invention, both as to its structure and operation can best be understood by referring to the following description with reference to the accompanying drawings in which:

FIG. 1 is a general block diagram of a system in which the present invention can be implemented;

FIG. 2 is a block diagram illustration of Portal server 102 depicted in FIG. 1;

FIG. 3 is a block diagram illustration of Remote Portal server 106 depicted in FIG. 1;

FIG. 4 is a block diagram illustration of Communication devices 114 depicted in FIG. 1;

FIG. 5 is a flow diagram which depicts the functions performed by the method of incorporating and providing RPS Modules to a Portal server 102 according to the present invention;

FIG. 6 is a flow diagram which depicts the functions performed by the method of eliminating RPS Modules from a Portal server 102 according to the present invention;

FIG. 7 is a flow diagram which depicts the functions performed by the method of adding RPS Modules into a portal according to the present invention;

FIG. 8 is a flow diagram which depicts the functions performed by the method of removing RPS Modules from a portal according to the present invention;

Fig. 9 is a flow diagram which depicts the functions performed by the method of enabling RPS Modules; and

Fig 10 is a flow diagram which depicts the functions performed by the method of disabling deployed RPS Modules.

#### **DETAILED DESCRIPTION OF THE INVENTION**

To facilitate an understanding of the present invention, it is described more fully hereinafter with reference to specific implementations of the accompanying

drawings that show the preferred embodiments of the invention. This invention, however, may be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Appropriately, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey  
5 the scope of the invention. As will be appreciated by one having skill in the art, the present invention may be embodied as a method, data processing system and computer program product.

The software programs that underlie the invention can be coded in different languages executable with different platforms. In the description that follows,  
10 examples of the invention are described in the context of web sites or service that employ Java Server Pages (JSP) or Active Server Pages (ASP). It will be appreciated, however, that the principles that underlie the invention can be implemented with other types of computer software object technologies as well. Furthermore, the present invention can be embodied as a computer program product stored on a computer-  
15 readable storage medium. Any suitable computer-readable medium may be utilized including hard disks, CD-ROMs, floppy disks, optical storage devices, magnetic storage devices, etc.

A general depiction of a system in which the present invention can be implemented is illustrated in FIG. 1. In a preferred embodiment of the present  
20 invention, system 100 depicted in FIG. 1 includes communication devices 114, Portal server 102, Remote Portal Service server 106 and network 104. In another embodiment of the present invention, a RPS system 100 includes communication

devices 114, Portal server 102, Remote Portal Service server 106, network 104 and Remote Portal Service Directory server 110. In another embodiment of the present invention, a RPS system 100 includes communication devices 114, Portal server 102, Remote Portal Service server 106, network 104 and Remote Portal Service Reference  
5 Directory server 112.

The RPS system 100 may transmit, using network 104, any combination of voice, video and/or data between devices. In the preferred embodiment of the invention, network 104 transmits data between computers 114, 102 and 106. In another embodiment of the invention, network 104 transmits data between  
10 communication devices 114, Portal server 102, Remote Portal Service server 106 and Remote Portal Service Directory server 110. In another embodiment of the present invention, network 104 transmits data between communication devices 114, Portal server 102, Remote Portal Service server 106 and Remote Portal Service Reference Directory server 112.

15 The RPS system 100 can provide a RPS Service. The method of providing a RPS Service includes enabling, disabling, incorporating, eliminating, adding and removing a RPS Service. Communication devices 114 may be any apparatus from which, and to which, any combination of voice, video and/or data can be transmitted over a network, such as network 104. Communication devices 114 can include  
20 computers 114a, web access devices 114b, workstations 114c, telecommunications devices 114d, and the like. Communications devices 114 may perform the function of accessing Portal server 102 and viewing informational content as provided by a RPS

Service exposed in a portal to a user as a module. Communications devices 114 may be used by end-users of Portal server 102, such as customers.

Portal server 102 can be any computer that stores application programs and data shared by users of network 104 and that uses libraries, such as Java libraries. In the preferred embodiment of the present invention, Portal server 102 supports Java Server Pages (JSP). In another embodiment of the present invention, Portal server 102 supports Active Server Pages (ASP). Portal server 102 may perform the functions of incorporating, eliminating, adding and removing a RPS Service. An individual or a number of individuals, who are responsible for managing Portal sever 102, may incorporate, eliminate, add and remove a RPS Service. The RPS server 106 may be any computer that stores application programs and data shared by users of network 104. The RPS server 106 may perform the functions of enabling and disabling a RPS Service. An individual or individuals responsible for managing RPS Services may use RPS server 106. The RPS Directory server 110 may perform the functions of locating and listing of RPS Services. The RPS Directory server 110 may be any computer that stores application programs and data shared by users of network 104. The data may include a listing of RPS services available throughout network 104. The RPS Reference Directory server 112 may perform the functions of locating and listing available RPS Services as well as indicating invalid information.

Communication devices 114, RPS server 106 and Portal server 102 may connect to one another by means of a suitable communications network 104. Communication network 104 may be a local area network, a wide area network, the

Internet, a wireless network, or the like. The network 104 may transfer information between Portal server 102, RPS server 106 and communications devices 114. The information transferred may include any combination of voice, video and/or data. Network 104 can be implemented as a wireless network or a wired network.

5           FIG. 2 is a block diagram illustration of Portal server 102. The Portal server 102 may include a central processing unit (CPU) 200, connected by a bus 208, to RAM 202, ROM 204, system memory 210, data store 212 and network interface 206. Network interface 206 is connected to network 104 for communication with RPS server 106 and communication devices 114. Portal server 102 may perform the functions of incorporating, eliminating, adding and removing a RPS Service. The Portal server 102 may connect to other network resources, for example to acquire information from the Internet or an intranet.

10           In the exemplary embodiment shown in FIG. 2, the various components of the Portal server 102 communicate through a system bus 208 or similar architecture. Accordingly, systems memory 210 is disposed in communication with CPU 200 and data storage 212 through bus 208. Typically, CPU 200 is a microprocessor, such as an INTEL PENTIUM® or AMD® processor, but may also be any processor that executes computer program instructions. Systems memory 210 is the workspace from which all program execution and data processing takes place for Portal server 102. Systems memory 210 includes Portal Server Software 216 and Operating System 214.

Operating System 214 provides overall system functionality. Operating System 214 is the program that, after being initially loaded into the computer, manages all the other programs in Portal server 102. Operating System 214 finds data and delivers it to Portal Server Software 216. Conversely, when Portal Server Software 216 is ready to output, the Operating System 214 transfers the data from Portal Server Software 216 to the appropriate destination. Operating System 214 is responsible for the central management of all devices. Operating System 214 calls on one or more drivers for input and output, and the drivers communicate with the corresponding hardware devices.

Portal Server Software 216 performs the functions that are implemented by Portal server 102 using a library of object-oriented classes 218. The classes may be those in the Java programming language developed by Sun Microsystems, also stored in systems memory 210. The classes 218 enable access to various databases, web servers, scripting environments and mail services. The classes 218 can provide connection to other servers. The classes 218 use other resources as needed, including a data store which provides object persistence via a suitable database interface. In the preferred embodiment of the invention, this connection may be provided by a JDBC interface over a SQL database. In another embodiment based upon an LDAP environment, user management can be provided via JNDI over LDAP.

In a preferred embodiment, Portal Server Software 216 is implemented using object-oriented technology, and thus uses objects as building blocks. Objects are software components designed to work together at runtime without any prior linking

or precompilation as a group. The objects represent actors within an overall system. A software component is a module that contains the data as well as the data structure and functions that manipulate the data.

Portal Server Software 216 may provide for the dynamic construction and maintenance of an initial set of views, or set of front pages (hereinafter “portals”), that includes a plurality of modules. Modules are objects that encapsulate a specific bound portion of content at a network address for administration as a unit. Modules follow a singleton design pattern such that only one instance of a module is maintained for the lifetime of a server session. Each module represents a resource that can be accessed by the user through the portal. Some of the modules can be implemented on Portal server 102, whereas other modules may be implemented on a remote host, such RPS server 106. Interaction with a module by end-users using communication devices 114 accesses the information or services provided by the module.

Portal Server Software 216 may implement a RPS Browser Module, a RPS Proxy, and RPS Service Implementation. A RPS Browser Module serves as an interface for performing administration. The RPS Browser Module may be provided as a graphical user interface having objects selectable by an administrator. A RPS Browser Module may instantiate a RPS Proxy. A RPS Proxy encapsulates and supports the use of a RPS Service implementation. A RPS Service Implementation can execute the behavior necessary to satisfy the RPS Protocol. The RPS Protocol is the communications protocol used between RPS Server Software and Portal Server Software 216 to exchange information. Any number of protocols can be used that



allows for the exchange of information over network 104 using network interface 206. Network interface 206 may enable the communication to be transmitted and received over a network 104.

In the preferred embodiment of the present invention, the RPS Protocol is implemented in XLM over Simple Object Access Protocol (hereinafter “SOAP”). SOAP is a lightweight protocol for exchanging information in a decentralized, distributed environment. It is an XML based protocol that includes three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols. However, in the preferred embodiment of the present invention, SOAP is used in combination with HTTP and HTTP Extension Framework.

SOAP does not itself define any application semantics such as a programming model or implementation specific semantics. SOAP defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC.

Data Storage 212 may be implemented in any number of forms. For example, data storage 212 may be a relational database. Data Storage 212 may store data such as documents, folders and web content. The data may be stored as a data structure, such as files. Data Storage 212 may cache retrieved web content and uncache timed out web content.

FIG. 3 is a block diagram illustration of Remote Portal server 106. The Remote Portal server 106 may include a central processing unit (CPU) 300, connected by a bus 308 to RAM 302, ROM 304, system memory 310, data storage 312 and network interface 306 which is connected to network 104 for communication with Portal server 102 and communication devices 114. The Remote Portal server 106 may perform the function of enabling and disabling a RPS Service for addition onto a portal provided by Portal server 102. The Remote Portal server 106 can connect to other network resources, for example to acquire information from the Internet or an intranet.

As shown, the various components of the Remote Portal server 106 communicate through a system bus 308 or similar architecture. Accordingly, systems memory 310 is disposed in communication with CPU 300 through bus 308. Typically, CPU 300 is a microprocessor, such as an INTEL PENTIUM® processor, but may also be any processor that executes computer program instructions. Systems memory 310 is the workspace from which all program execution and data processing takes place for Remote Portal server 106. Systems memory 310 includes RPS Server Software 312 and Operating System 314.

Operating System 314 provides overall system functionality. Operating system 314 is the program that, after being initially loaded into the computer, manages all the other programs in server 102. The Operating System 314 finds data and delivers it to RPS Server Software 312. Conversely, when RPS Server Software 312 is ready to output, the operating system 314 transfers the data from RPS Server Software 312 to

the appropriate destination. The Operating System 314 is responsible for the central management of all devices. The Operating System 314 calls on one or more drivers for input and output, and the drivers communicate with the corresponding hardware devices.

5 RPS Server Software 312 provide the functions that are implemented by RPS server 106. RPS Server Software 312 implements object-oriented technology and thus uses objects as building blocks. The objects supported by RPS Server Software 312 include RPS Configuration Descriptor Source, RPS Service Implementation and RPS Module.

10 RPS Configuration Descriptor Source may be implemented by RPS Server Software 312. RPS Configuration Descriptor Source supplies the RPS Configuration Descriptor for a RPS Service Implementation. In the preferred embodiment of the present invention, A Configuration Descriptor is provided as an XML file. A Configuration Descriptor provides configuration information pertaining to the RPS  
15 Service Implementation of a RPS Module. A RPS Module corresponds to a RPS Service hosted by RPS Server 106. RPS Module may be implemented as a PortalBean. The PortalBean is a data structure that may include a descriptor file and view information. The PortalBean may be provided as an XML file. The descriptor file can include description and property information for a RPS Module. The view  
20 information may be used to dynamically generate a view. The RPS Module uses the RPS Protocol to communicate with Portal Server Software 216. Network interface

306 may enable communications between RPS Server Software 312 and Portal Server Software 216 to be transmitted and received over a network 104.

FIG. 4 is a block diagram illustration of Communications devices 114. The Communications devices 114 may include CPU 400, connected by a bus 408 to RAM 402, ROM 404, network interface 406, and system memory 410. Communications device 114 can also include input device interface 412, and display interface 414. Input device interface 412 enables interaction with and execution of instruction by communication device 114 as directed by a user. Display interface 414 can display information generated for output by communication device 114 as provided by Portal server 102. The information may include portals that have a plurality of modules.

As shown, the various components of the communication device 114 communicate through a bus 408 or similar architecture. Accordingly, systems memory 410 is disposed in communication with CPU 400 through bus 408. Systems memory 410 includes Browser Program 416 and Operating System 418.

Operating system 418 provides overall system functionality. Browser Program 416 is computer program instructions executed by CPU 400. The browser program 308 enables the information transmitted from Portal server 102 to be conveyed to a user in a manner that can be understood by a user of communications devices 114. The browser 308 serves as a front end to the World Wide Web on the Internet.

FIG. 5 is a flow diagram which depicts the functions performed by the method of incorporating and providing RPS Modules to a Portal server 102 according to the

present invention. As used herein the term remote services is meant to convey applications, content and services hosted on a computer remote from a computer implementing a portal. The remote services are exposed in the portal as RPS Modules without the installation of software specific to the remote applications, content or services.

In Step 500, an external reference identifier is obtained. The external reference identifier may represent a RPS Service Implementation for a RPS Module. In an embodiment of the present invention, the external reference identifier may be listed on a directory. The directory may be hosted by RPS Directory sever 110 or RPS Reference Directory server 112. A RPS Browser Module may provide the directory as well as other object useful for the administration of a portal. A RPS Browser Module may permit instantiation of a RPS Proxy. A RPS Proxy may be instantiated in response to manipulation of an external reference identifier. A RPS Proxy encapsulate and support the use of a RPS Service Implementation. An individual responsible for managing a portal may obtain and manipulate the external reference identifier. For example, the individual may be an administrator or a group of administrators for Portal server 102.

In Step 502, Portal server 102 obtains a RPS Configuration Descriptor for a RPS Service Implementation of a RPS Module on RPS server 106. The RPS Configuration Descriptor may be transmitted over network 104 from RPS server 106. In the preferred embodiment of the present invention, the RPS Configuration Descriptor may be obtained by issuing a Configuration Descriptor Request from

Portal server 102 over network 104 to RPS sever 106. A Configuration Descriptor Response provides the configuration information pertaining to the RPS implementation of the RPS Module represented by the external identifier. The Configuration Descriptor Response is transmitted from RPS server 106 over network 104 to Portal server 102. The Configuration response and request are provided using the RPS Protocol.

In Step 504, the RPS Module may be added into Portal server 102. In the preferred embodiment of the present invention, the RPS Module is added by selecting the external reference identifier. In Step 506, a PortalBean Descriptor for the RPS Module is obtained from RPS server 106. The PortalBean Descriptor obtained is for the RPS Module selected by the user Step 504. In the preferred embodiment of the present invention, the PortalBean Descriptor is generated and based on the RPS Configuration Descriptor. In an embodiment of the present invention, the PortalBean Descriptor for the RPS Module is pre-generated. The PortalBean Descriptor is transmitted from RPS server 106 over network 104 to Portal server 102.

In Step 508, the PortalBean Descriptor for the incorporated RPS Module is stored with other PortalBean Descriptors in the memory 310 of RPS Server 106. In Step 510, the set of PortalBean Descriptors that includes the PortalBean Descriptor for the added RPS Module may be used to generated a palette used by the Portal server 102.

FIG. 6 is a flow diagram which depicts the functions performed by the method of eliminating RPS Modules from a Portal server 102 according to the present

invention. The RPS Module corresponds to a RPS Service incorporated onto Portal server 102. The RPS Module may be represented on the palette of Portal server 102.

In Step 600, a palette that includes RPS Modules incorporated onto Portal server 102 is generated. The palette may be display on an interface. An individual responsible for managing the Portal server 102, such as an administrator, may use the interface to perform administration on Portal server 102. In Step 602, a RPS Module is selected. The selected RPS Module corresponds to the RPS Module to be removed from Portal server 102. In Step 604, the PortalBean for the selected RPS Module is deleted. [ In Step 606, the set of PortalBean Descriptors, except the PortalBean Descriptor for the removed RPS Module, may be used to generated a palette used by the Portal server 102.

FIG. 7 is a flow diagram which depicts the functions performed by the method of adding RPS Modules into a portal according to the present invention. The RPS Module corresponds to the remote portal service incorporated onto Portal server 102. The RPS Module may be included into a portal of Portal server 102.

In Step 700, a palette that includes RPS Modules incorporated onto Portal server 102 is generated. The palette may be display on an interface. The palette includes all RPS Modules incorporated onto Portal server 102. An individual responsible for managing the Portal server 102, such as an administrator, may use the interface to perform administration on Portal server 102. In Step 702, a RPS Module is selected. In Step 704, Portal server 102 obtains a RPS Configuration Descriptor. The RPS Configuration Descriptor is obtained from the selected RPS Module. In Step

706, the RPS Configuration Descriptor is stored in memory or RPS Server 106. The RPS Configuration Descriptor may be used to connect to the RPS Module.

FIG. 8 is a flow diagram which depicts the functions performed by the method of removing RPS Modules from a portal according to the present invention. The RPS Module corresponds to the remote portal service incorporated onto Portal server 102. The RPS Module may be included into a portal of Portal server 102.

In step 800, all RPS modules incorporated onto portal server 102 may be provided. In the preferred embodiment, RPS modules are displayed on a palette. The palette may be displayed on an interface. In step 802, the RPS module to be removed may be designated. In the preferred embodiment, the RPS module to be removed is selected from a palette. The selection may be made by an individual responsible for managing portal server 102. In step 804, descriptor information related to the RPS module is deleted. In the preferred embodiment, the RPS configuration descriptor is deleted from memory of RPS Server 106. The deletion of the RPS configuration descriptor prevents portal server 102 from connecting to a RPS module on RPS server 106.

Fig. 9 is a flow diagram which depicts the functions performed by the method of enabling a RPS Modules. The RPS module may be hosted on RPS server 106. The RPS Module may be services of third party content and applications providers.

In step 900, Module information may be provided. In the preferred embodiment, the Module information includes external reference data and type data.



The external reference data and type data is related to a RPS Module to be deployed. The deployed RPS module may be incorporated and used by portal server 102. In an embodiment, Module information includes initialization data. The initialization data may be parameters that specifies the configuration specification necessary for proper operation of the module. The data may be provided by an individual responsible for managing RPS Server 106. For example, the individual may be an administrator of RPS server 106. The deployed RPS Module may be provided on a directory. In an embodiment, the RPS Module is provided on a RPS Directory server 110. In another embodiment, the RPS Module is provided on a RPS Reference Directory server 112. The server 110 and 112 may be accessed for the review and incorporation of the RPS module. In step 902, RPS server 106 may instantiate the RPS module. In step 904, RPS server 106 initializes the RPS module. In step 906, the RPS module is deployed for use.

Fig 10 is a flow diagram which depicts the functions performed by the method of disabling a deployed RPS module. The RPS module may be hosted on RPS server 106. The RPS module may be removed when the service represented by the RPS module is no longer appropriate for distribution and use.

In step 1000, the existing RPS modules currently deployed may be provided. In the preferred embodiment, the existing deployed RPS modules may be listed on a display. In step 1002, a RPS module may be designated for removal. In the preferred embodiment, the designation occurs though the selection of the RPS module is to be removed. The selection may be made by an individual responsible for the

management of RPS server 106. For example an administrator of RPS server 106. In an embodiment, several RPS modules are selectable for simultaneous removal. In step 1004, module information is canceled. In the preferred embodiment the module information includes external reference data and type data. In step 1006, the RPS module is terminated. A terminated RPS modules removes the RPS module from use. For example, the RPS module may be eliminated from directory listings.

The present invention is described below with reference to the illustration of the methods, apparatus, computer program product and method of doing business. It will be understood that each block of the flowchart illustrations of FIGS. 5-10, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These computer program instructions may be loaded onto a general-purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine. These computer program instructions, which execute on the computer or other programmable data processing apparatus, create means for implementing the functions specified in the flowchart block or blocks. These computer program instructions may be stored in a computer-readable memory to direct a computer or other programmable data processing apparatus to function in a particular manner, producing an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed, producing a computer implemented process, such

that the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks. In the preferred embodiment, these computer program instructions are written in an object oriented programming language, such as C, C++ and Java. It is to be appreciated, however, that these  
5 computer program instructions may be implemented in any of a wide variety of programming languages.

**CLAIMS**

1     What is claimed is:

2     1. A method for providing RPS modules, the method comprising:

3         enabling a Remote Portal Service (RPS) module; and

4         incorporating the RPS module into a networked portal server.

1     2. The method according to claim 1, further comprising generating content as defined

2     by the RPS module.

1     3. The method according to claim 2, wherein the generating of the content includes

2     determining an upgrade state.

1     4. The method according to claim 3, further comprising performing a modification.

1     5. The method according to claim 3, wherein the determining includes disabling the

2     RPS module.

1     6. The method according to claim 1, wherein the enabling of the RPS module

2     includes specifying an external reference identifier and a type identifier.

1     7. The method according to claim 6, wherein the enabling of the RPS module

2     includes specifying an initialization parameter.

1 8. The method according to claim 6, wherein the enabling of the RPS module  
2 includes specifying a permission.

1 9. The method according to claim 1, wherein the incorporating of the RPS module  
2 includes obtaining an external reference identifier

1 10. The method according to claim 9, wherein the obtaining of the external reference  
2 identifier includes accessing a listed RPS directory server.

1 11. The method according to claim 10, wherein the accessing of the RPS directory  
2 server includes selecting the RPS directory server.

1 12. The method according to claim 1, wherein the incorporating of the RPS module  
2 includes obtaining a RPS configuration descriptor.

1 13. The method according to claim 12, wherein the incorporating of the RPS module  
2 includes generating a first descriptor based on the RPS configuration descriptor.

1 14. The method according to claim 13, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 15. The method according to claim 1, wherein incorporating of the RPS module  
2 includes obtaining a first descriptor.

1 16. The method according to claim 15, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 17. The method according to claim 1, further comprising eliminating the RPS module  
2 from the networked portal server.

1 18. The method according to claim 17, wherein the eliminating of the RPS module  
2 includes deleting a first descriptor.

1 19. The method according to claim 1, further comprising disabling the RPS module.

1 20. The method according to claim 19, wherein the disabling of the RPS module  
2 includes deleting an external reference identifier.

1 21. The method according to claim 19, wherein the disabling includes selecting the  
2 RPS module.

1 22. The method according to claim 1, further comprising modifying an access  
2 permission.

1 23. The method according to claim 22, wherein the modifying of the access  
2 permission includes specifying a new permission setting.

1 24. The method according to claim 1, further comprising adding the RPS module to a  
2 portal.

1 25. The method according to claim 24, wherein the adding of the RPS module to the  
2 portal includes obtaining a RPS configuration descriptor.

1 26. The method according to claim 25, wherein the adding of the RPS module to the  
2 portal includes storing the RPS configuration descriptor.

1 27. The method according to claim 26, wherein the adding of the RPS module to the  
2 portal includes connecting to the RPS module based on the RPS configuration  
3 descriptor.

1 28. The method according to claim 24, further comprising removing the RPS module  
2 from the portal.

1 29. The method according to claim 28, wherein the removing of the RPS module from the  
2 portal includes deleting a RPS configuration descriptor.

1 30. A system for providing RPS module, the system comprising:

2 a first server for enabling a Remote Portal Service (RPS) module;

3 a network; and

4 a second server, connected to the first server and the network, for incorporating the

5 RPS module.

1 31. The system according to claim 30, further comprising a computer, connected to the

2 network, for generating content as defined by the RPS module.

1 32. The system according to claim 31, wherein the generating of content includes

2 determining an upgrade state.

1 33. The system according to claim 32, further comprising performing a modification

2 to the RPS module.

1 34. The system according to claim 32, wherein the determining includes disabling the

2 RPS module.

1 35. The system according to claim 30, wherein the enabling of the RPS module

2 includes obtaining an external reference identifier and a type identifier.



1 36. The system according to claim 35, wherein the enabling of the RPS module  
2 includes obtaining an initialization parameter.

1 37. The system according to claim 35, wherein the enabling of the RPS module  
2 includes obtaining a permission.

1 38. The system according to claim 30, wherein the incorporating of the RPS module  
2 includes obtaining an external reference identifier.

1 39. The system according to claim 38, wherein the obtaining of the external reference  
2 identifier includes accessing a directory.

1 40. The system according to claim 39, wherein the accessing of the directory includes  
2 selecting the directory.

1 41. The system according to claim 30, wherein the incorporating of the RPS module  
2 includes obtaining a RPS configuration descriptor.

1 42. The system according to claim 41, wherein the incorporating of the RPS module  
2 includes the second sever generating a first (i.e java bean) descriptor based on the RPS  
3 configuration descriptor (text file).

1 43. The system according to claim 42, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 44. The system according to claim 30, wherein incorporating of the RPS module  
2 includes obtaining a first descriptor.

1 45. The system according to claim 44, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 46. The system according to claim 30, further comprising the second server for  
2 eliminating the RPS module from the networked portal server.

1 47. The system according to claim 46, wherein the eliminating of the RPS module  
2 includes deleting a first descriptor.

1 48. The system according to claim 30, further comprising the first server for disabling  
2 the RPS module.

1 49. The system according to claim 48, wherein the disabling of the RPS module  
2 includes deleting an external reference identifier.

- 1 50. The system according to claim 48, wherein the disabling includes selecting the  
2 RPS module.
- 1 51. The system according to claim 30, further comprising the first server for  
2 modifying an access permission.
- 1 52. The system according to claim 51, wherein the modifying of the access permission  
2 includes specifying a new permission setting.
- 1 53. The system according to claim 30, further comprising the second server for adding  
2 the RPS module to a portal.
- 1 54. The system according to claim 53, wherein the adding of the RPS module to the  
2 portal includes obtaining a RPS configuration descriptor.
- 1 55. The system according to claim 54, wherein the adding of the RPS module to the  
2 portal includes storing the RPS configuration descriptor.
- 1 56. The system according to claim 55, wherein the adding of the RPS module to the  
2 portal includes the second server for connecting to the RPS module based on the RPS  
3 configuration descriptor.

57. The system according to claim 53, further comprising the second server for removing the RPS module from the portal.

58. The system according to claim 57, wherein the removing of the RPS module from the portal includes deleting a RPS configuration descriptor.

59. A device for managing the deployment of RPS modules, the device comprising:  
a network interface exchanging information with a network;  
a memory storing program instructions; and  
a processor, coupled to the memory and the network interface, for enabling a Remote Portal Service module.

60. The device according to claim 59, wherein the enabling of the RPS module includes obtaining an external reference identifier and a type identifier.

61. The device according to claim 60, wherein the enabling of the RPS module includes obtaining an initialization parameter.

62. The device according to claim 60, wherein the enabling of the RPS module includes obtaining a permission.

63. The device according to claim 59, further comprising the processor for disabling the RPS module.

64. The device according to claim 63, wherein the disabling of the RPS module includes deleting an external reference identifier.

65. The device according to claim 63, wherein the disabling includes selecting the RPS module.

66. The device according to claim 59, further comprising the processor for modifying an access permission.

67. The device according to claim 66, wherein the modifying of the access permission includes obtaining a new permission setting.

68. The device according to claim 59, wherein the exchanging of information is performed based on a XML over SOAP protocol.

69. A device for managing deployed RPS modules, the device comprising:  
a network interface exchanging information with a network;  
a memory storing program instructions; and  
a processor, connected to the network interface and memory, for incorporating the RPS module.

1 70. The device according to claim 69, wherein the incorporating of the RPS module  
2 includes obtaining an external reference identifier.

1 71. The device according to claim 70, wherein the obtaining of the external reference  
2 identifier includes accessing a directory.

1 72. The device according to claim 71, wherein the accessing of the directory includes  
2 selecting the directory.

1 73. The device according to claim 69, wherein the incorporating of the RPS module  
2 includes obtaining a RPS configuration descriptor.

1 74. The device according to claim 73, wherein the incorporating of the RPS module  
2 includes generating a first descriptor based on the RPS configuration descriptor.

1 75. The device according to claim 74, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 76. The device according to claim 69, wherein incorporating of the RPS module  
2 includes obtaining a first descriptor.

1 77. The device according to claim 76, wherein the incorporating of the RPS module  
2 includes storing the first descriptor.

1 78. The device according to claim 69, further comprising the processor eliminating the  
2 RPS module from the networked portal server.

1 79. The device according to claim 78, wherein the eliminating of the RPS module  
2 includes deleting a first descriptor.

1 80. The device according to claim 69, further comprising the processor adding the  
2 RPS module to a portal.

1 81. The device according to claim 80, wherein the adding of the RPS module to the  
2 portal includes obtaining a RPS configuration descriptor.

1 82. The device according to claim 81, wherein the adding of the RPS module to the  
2 portal includes storing the RPS configuration descriptor.

1 83. The device according to claim 81, wherein the adding of the RPS module to the  
2 portal includes connecting to the RPS module based on the RPS configuration  
3 descriptor.

1 84. The device according to claim 69, further comprising the processor removing the  
2 RPS module from the portal.

1 85. The device according to claim 84, wherein the removing of the RPS module from the  
2 portal includes deleting a RPS configuration descriptor.

1 86. The device according to claim 69, wherein the exchanging of information is  
2 performed based on a XML over SOAP protocol.

3  
4 87. A computer program product comprising:

5 a computer useable medium including computer program instructions for managing the  
6 deployment of RPS Services; and  
7 computer program instructions for enabling a Remote Portal Service module.

1 88. The computer program product according to claim 87, wherein the enabling of the  
2 RPS module includes obtaining an external reference identifier and a type identifier.

1 89. The computer program product according to claim 87, wherein the enabling of the  
2 RPS module includes obtaining an initialization parameter.

1 90. The computer program product according to claim 87, wherein the enabling of the  
2 RPS module includes obtaining a permission.



1 91. The computer program product according to claim 87, further comprising  
2 computer program instructions for disabling the RPS module.

1 92. The computer program product according to claim 91, wherein the disabling of the  
2 RPS module includes deleting an external reference identifier.

1 93. The computer program product according to claim 91, wherein the disabling  
2 includes selecting the RPS module.

1 94. The computer program product according to claim 87, further comprising  
2 computer program instructions for modifying an access permission.

1 95. The computer program product according to claim 94, wherein the modifying of  
2 the access permission includes obtaining a new permission setting.

1 96. A computer program product comprising:  
2 a computer useable medium including computer program instructions for managing  
3 deployed RPS Services; and  
4 computer program instructions for incorporating the RPS module.

1 97. The computer program product according to claim 96, wherein the incorporating  
2 of the RPS module includes obtaining an external reference identifier.

1 98. The computer program product according to claim 97, wherein the obtaining of the  
2 external reference identifier includes accessing a directory.

1 99. The computer program product according to claim 98, wherein the accessing of the  
2 directory includes selecting the directory.

1 100. The computer program product according to claim 96, wherein the  
2 incorporating of the RPS module includes obtaining a RPS configuration descriptor.

1 101. The computer program product according to claim 100, wherein the  
2 incorporating of the RPS module includes the second sever generating a first  
3 descriptor based on the RPS configuration descriptor.

1 102. The computer program product according to claim 101, wherein the  
2 incorporating of the RPS module includes storing the first descriptor.

1 103. The computer program product according to claim 96, wherein incorporating of  
2 the RPS module includes obtaining a first descriptor.

1 104. The computer program product according to claim 103, wherein the  
2 incorporating of the RPS module includes storing the first descriptor.

1 105. The computer program product according to claim 96, further comprising  
2 computer program instructions for eliminating the RPS module from the networked  
3 portal server.

1 106. The computer program product according to claim 105, wherein the  
2 eliminating of the RPS module includes deleting a first descriptor.

1 107. The computer program product according to claim 96, further comprising  
2 computer program instructions for adding the RPS module to a portal.

1 108. The computer program product according to claim 107, wherein the adding of  
2 the RPS module to the portal includes obtaining a RPS configuration descriptor.

1 109. The computer program product according to claim 108, wherein the adding of  
2 the RPS module to the portal includes storing the RPS configuration descriptor.

1 110. The computer program product according to claim 108, wherein the adding of  
2 the RPS module to the portal includes connecting to the RPS module based on the  
3 RPS configuration descriptor.

1 111. The computer program product according to claim 96, further comprising  
2 computer program instructions for removing the RPS module from the portal.

- 1 112. The computer program product according to claim 111, wherein the removing of the
- 2 RPS module from the portal includes deleting a RPS configuration descriptor.

**ABSTRACT OF THE INVENTION**

The present invention is a method for providing Remote Portal Services (RPS). RPS is a system for easily and quickly incorporating remote applications (RPS Services) into a portal. RPS Services may be enabled making the RPS Service available for incorporation onto a server implementing a portal. Enabled RPS Services may be disabled discontinuing the availability of the RPS Service for incorporation onto a server implementing a portal. RPS Services may be incorporated onto and removed from a server implementing a portal server. RPS Services may be added to and removed from a portal implemented on a server. RPS Services are exposed in the portal as modules, without installing, on the server implementing the portal, any software specific to a RPS Service. Enabled RPS Services may be listed on a directory that may be searched and browsed. The present invention may protect the exposure of data stored on servers during communication between servers.

FIG. 1

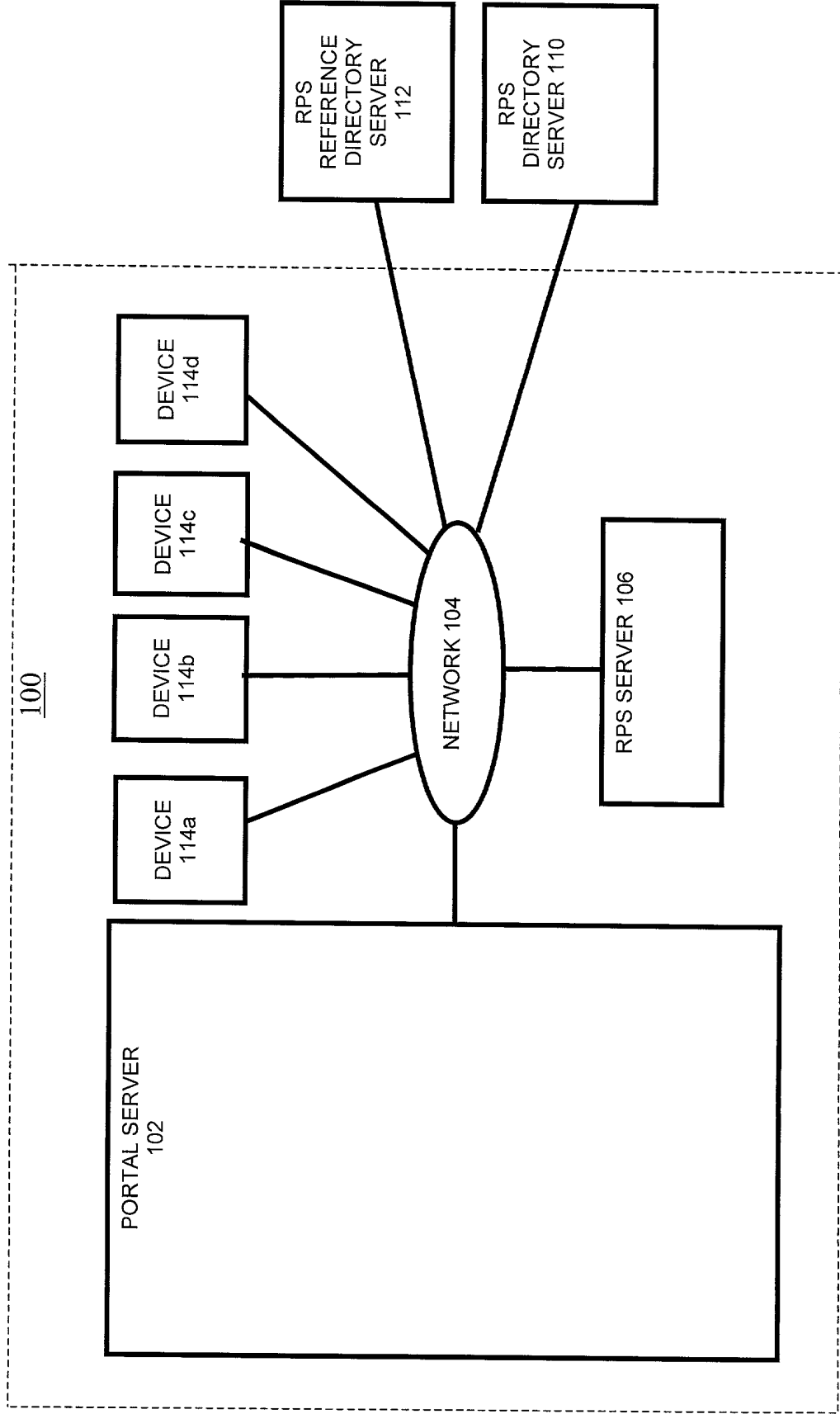


FIG. 2

PORTAL SERVER

102

MEMORY  
210

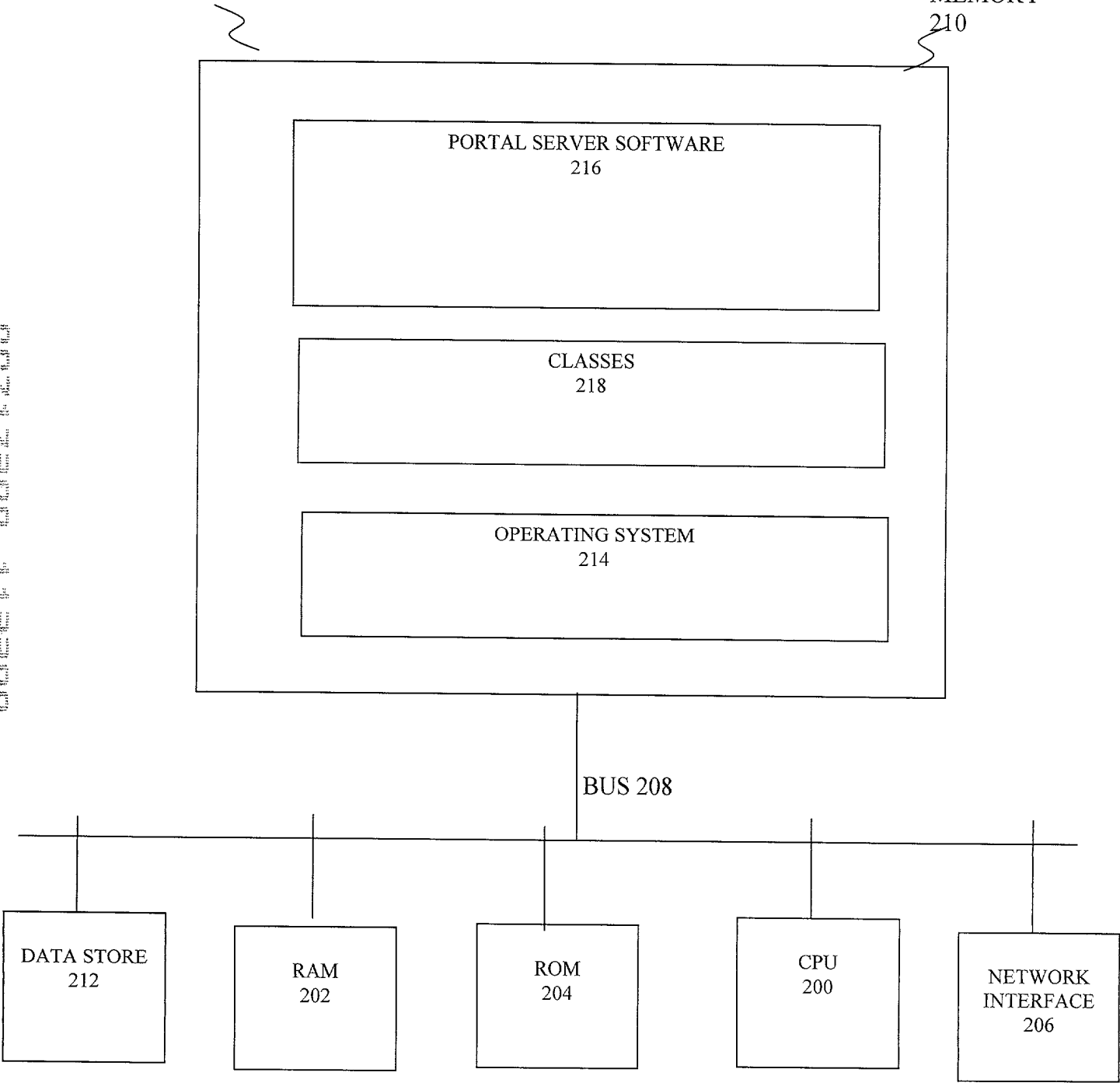


FIG. 3

REMOTE PORTAL SERVER 106

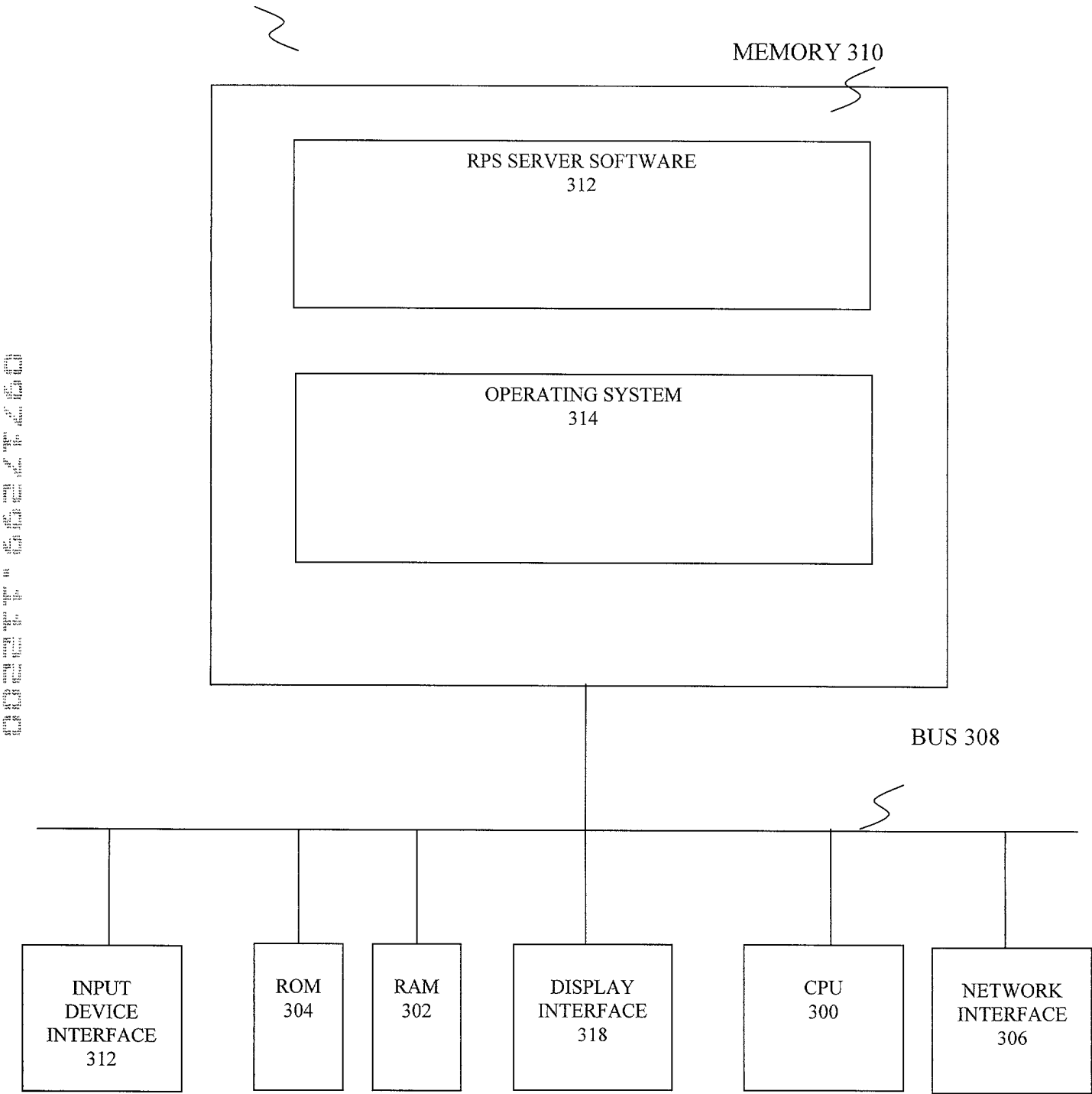
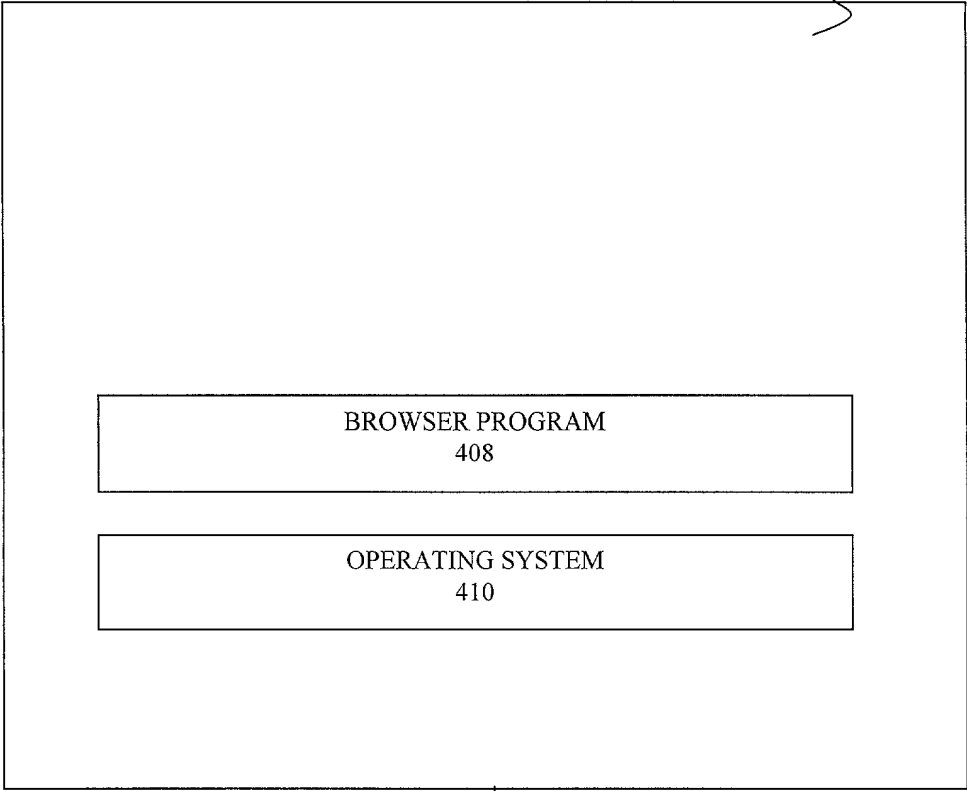




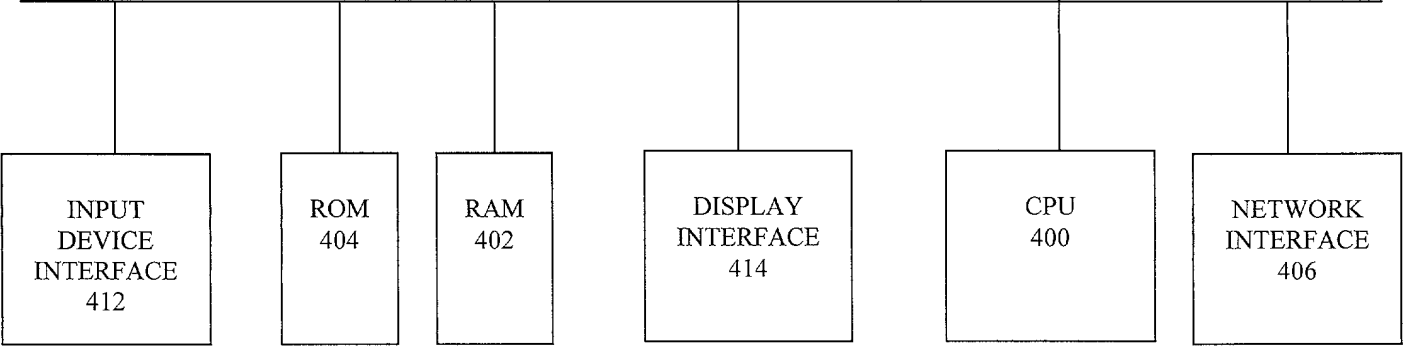
FIG. 4

COMMUNICATION DEVICE 114

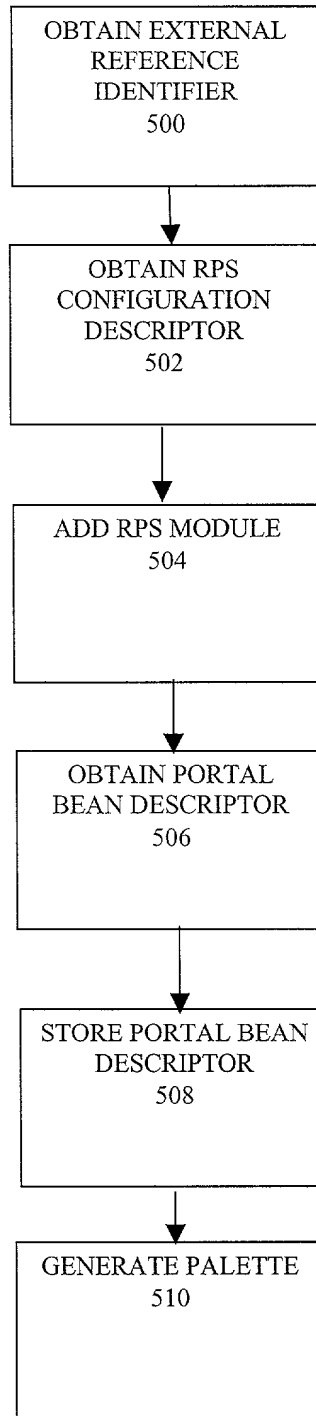
MEMORY 410



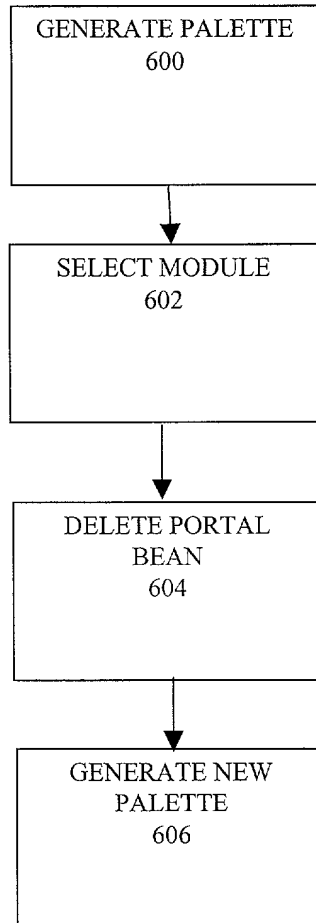
BUS 408



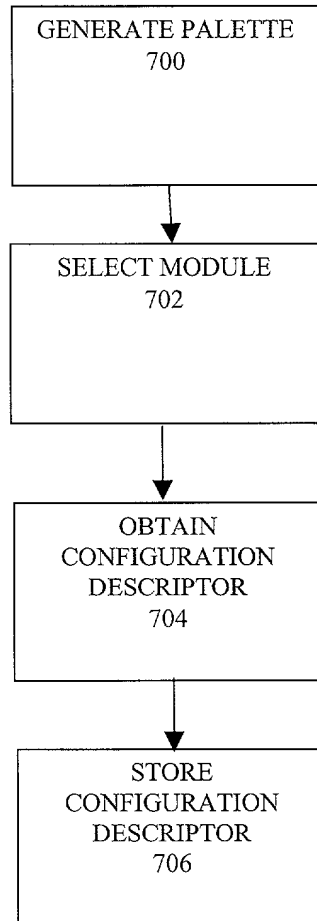
**FIG. 5**



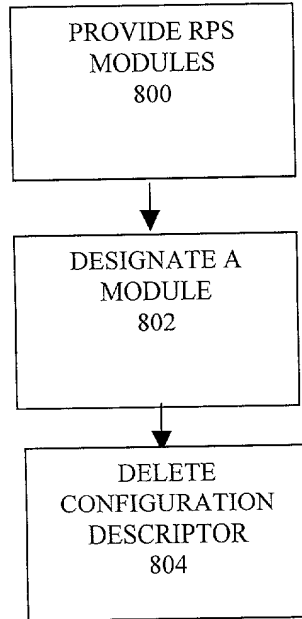
**FIG. 6**



**FIG. 7**



**FIG. 8**



**FIG. 9**

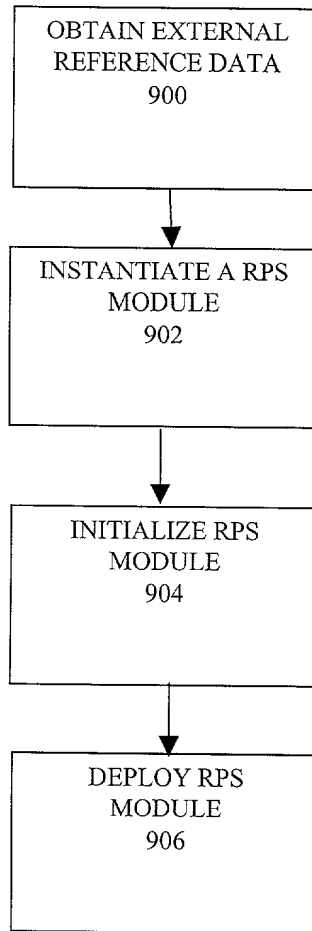


FIG. 10

